SHIP PRODUCTION COMMITTEE
FACILITIES AND ENVIRONMENTAL EFFECTS
SURFACE PREPARATION AND COATINGS
DESIGN/PRODUCTION INTEGRATION
HUMAN RESOURCE INNOVATION
MARINE INDUSTRY STANDARDS
WELDING
INDUSTRIAL ENGINEERING
EDUCATION AND TRAINING

September 1982
NSRP 0009

THE NATIONAL SHIPBUILDING RESEARCH PROGRAM

Proceedings of the IREAPS Technical Symposium

Paper No. 9: HULSTRX: A U.S. Navy Structural Design Model

U.S. DEPARTMENT OF THE NAVY
CARDEROCK DIVISION,
NAVAL SURFACE WARFARE CENTER

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 1982**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **The National Shipbuilding Research Program, Proceedings of the IREAPS Technical Symposium Paper No. 9: HULSTRX: A U.S. Navy Structural Design Model**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center CD Code 2230 - Design Integration Tools Building 192 Room 128-9500 MacArthur Blvd Bethesda, MD 20817-5700**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

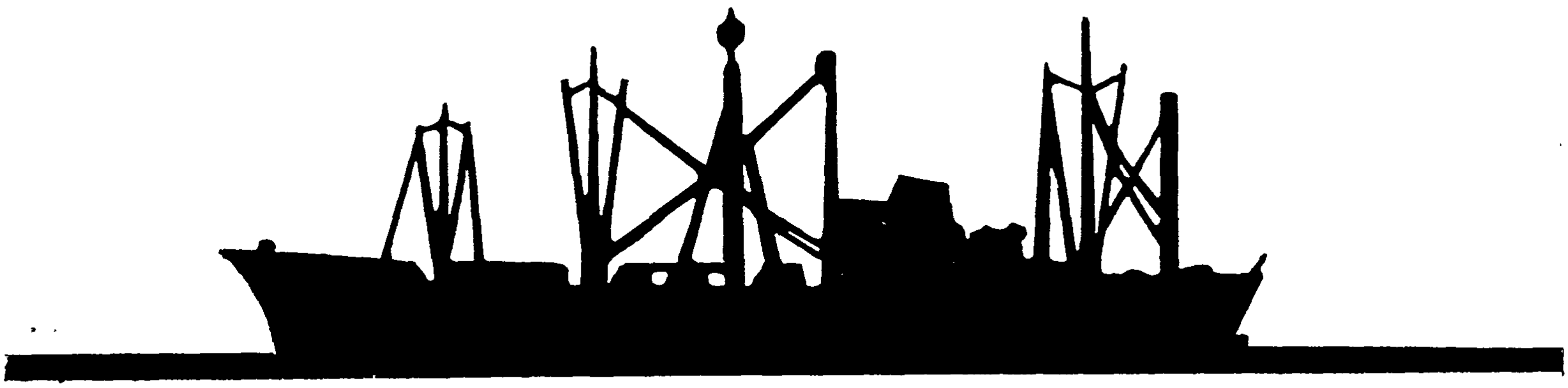
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**23**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

DISCLAIMER

Proceedings
IREAPS Technical Symposium
September 14-16-1982
San Diego, California

VOLUME I



INSTITUTE FOR RESEARCH AND ENGINEERING FOR AUTOMATION AND PRODUCTIVITY IN SHIPBUILDING

I R E A P S

HULSTRX: A U.S. NAVY STRUCTURAL DESIGN MODEL

Jon Gude
Designers & Planners Incorporated
Arlington, Virginia

Mr. Gude is a Naval Architect at Designers and Planners Inc, supervising software development. Mr. Gude has worked for ship design agents and shipyards in Norway and the United States. He has 6 years experience at Sun Ship in Chester, Pennsylvania working primarily on the Swedish STEERBEAR system. Mr. Gude also has extensive experience with AUTOKON system.

Mr. Gude received his B.Sc degree in naval architecture and shipbuilding from Kings College in University of Dusham, England. He also holds a MA degree in economics from University of Delaware.

Stephen H. Klomparens
Designers & Planners Incorporated
Arlington, Virginia

Mr. Klomparens is a Naval Architect at Designers & Planners Inc, and is engaged in development of computer aids for ship design. Mr. Klomparens has 6 years of experience with marine laboratory facilities conducting testing and development of conventional and advanced marine craft. Since 1977, he has been involved with naval and commercial ship design and with development of computer aided ship design tools.

Mr. Klomparens received his BSE degree in naval architecture and marine engineering from the University of Michigan, and his MS degree in computer science from the Johns Hopkins University.

HULSTRX- A U.S. NAVY STRUCTURAL DESIGN MODEL

by

Jon Gude
Designers & Planners, Inc.

Thomas P. Gallagher
Surface Ship Structural Branch, NAVSEA

Stephen H. Klomparens
Designers & Planners, Inc.

ABSTRACT

The concepts and status of the HULSTRX development along with a discussion of its utility in the rapidly changing U.S. Navy ship design environment are presented. HULSTRX provides for definition of ship structure at preliminary and contract design levels in terms of absolute geometric data and relational component data. Relational structural component definitions allow for quick design changes, and absolute geometric description of the hull surface can be independently defined. Application routines will enable designers to efficiently produce design drawings, various structural and geometric analyses, and interface with design or product models of other systems.

IREAPS PAPER ON HULSTRX - A U.S. NAVY STRUCTURAL DESIGN MODEL

HULSTRX - Present Status

The dual purpose of this paper is to report on the work that has been done in adding the interactive Structural Scantling File (SSF) to the Hull Stucture (HULSTRX) program and to outline the work to be done this fall in making the SSF an interactive design tool with graphics capability. In addition, the outlook of the U.S. Navy on the use of computer aided design tools will be reviewed, and some thoughts will be presented on the major considerations of a general data base design of the future.

Several papers have been presented on the HULSTRX progress during the last few years. See References 2, 3, and 4. These papers describe the background for the development of HULSTRX.

HULSTRX was last reported on in June of this year (Reference 4). The addition of the Ship Structural File to the HULSTRX programs has given the structural designer the capability to interactively add structures to the geometry of the ship, and to have structural drawings produced both on a graphics terminal as well as on a plotter.

The workings of the SSF program will be discussed a little later. The way in which it and the extensions to it interface with the existing HULSTRX files are shown on Fig. 1. SSF is essentially a stand-alone program that will create its own interactive database in which the structural details are stored. These details describe the structure with respect to type, material and extents. The traces on which they are to be placed are described in geometrical terms on the Design Geometry Library (DGL) file created by HULSTRX.

The next step in the direction of the generation of drawings is to add capabilities to the SSF Program so that drawings can be generated. Such drawing will first be displayed on a graphics screen and later on on a plotter or drafting machine. This effort is under way and will be completed by the end of the year. (Fig. 2) It is recognized that HULSTRX also should be converted to an interactive design tool. Such an effort may be started next year.

HULSTRX is the structural portion of Computer Supported Design (CSD). The CSD system is a set of linked individual computer programs which assist the cognizant engineer in developing a ship design. The shipbuilding industry is in a changing posture. The change is not just rhetoric, but a continuing effort. The needs of the industry should cover the entire spectrum of ship design, construction, overhaul, and repair. These needs have been recognized for many years but the computer hardware and software, and people that are eager to make the change, have recently became available to the industry in the numbers needed to exercise the change. The method of design must change and the time required to complete a design must be reduced. Since the use of the computer was accepted as a tool of the industry, the way to control it had been somewhat lost. Manual methods allowed three (3) to eight (8) design studies for a single ship design to be firmed up. With the computer, the studies go into the hundreds and the design process has grown, not decreased. Management of a computer-assisted design is a must if the real value of computer assistance is to be realized.

Design for design's sake should no longer be tolerated with computer-assisted designs. The process must be a continuous effort and build from the previous work done and not start over for each phase of design and construction. The manual method gave us little option, but that has changed. The ability to work from structured programs with a data base allows for true standards. The control of time, schedule, cost and quality can now be logically accomplished with a systematic approach.

The needs of the industry should be keyed to professional training. The people presently studying engineering and even many trades are exposed to computers and the mystique of how they work is gone. Home computers, video games, and TV graphics have set the tone of the country. The youth of the country are ready for a computerized industrial base and ship designers and builders should be ready.

HULSTRX is one of the tools being developed to meet the challenge. The system is being built on the strength of other programs and the needs of the user. HULSTRX has many requirements; no thought was given to develop a special stand alone system. HULSTRX as presently defined is about sixty-five (65) percent complete. Based on the original program plan, the present status should be ninety (90) percent complete. This is a true indication of how the needs of a system grows as user's.

HULSTRX is a relatively new program when one considers the time it takes to develop a system of this nature. The major goal is to computerize the process and to develop computer graphic drawings from preliminary through contract design. In-house designs that are done by the U.S. Navy without contractor support are done with engineers. The Surface Ship Structures Division has only three (3) types of people working in the office; engineers, engineers-in-training, and engineering aids. The engineers are full time employees that selected the Surface Ship Structures Division as their main area of employment. Engineers -in-training are the young engineers that are starting full time careers with the Navy. They spend four (4) weeks in each code as part of their exposure to the various discipline in the NAVSEA design area. Engineering aids are students, usually with one (1) to three (3) years college that work summers and long holidays. This limited description is given to show the need for a special system which will be able to carry a larger workload, thus reducing personnel needs.

Most people that work in the design area must realize that engineers do not like to draw drawings in the formal sense of the word. They will sketch structures, develop load diagrams, and spend numerous hours developing a finite element model but that all falls within the engineering arena.

HULSTRX took into account these user constraints into account in the planning and development. The basic development *was* the line definition of a shell expansion; this was selected as a keystone requirement and when accomplished would allow for the development of the simpler surfaces such as decks and bulkheads. The next phase was to develop the scantling file that is the subject of this paper and will be covered in detail later on.

Having a computer drafting program as support, *a* structural engineer should directly interface with his engineering tools. One of the major programs that is used by the Navy is the Structural Synthesis Design Program (SSDP). This program designs all longitudinal structures for Navy type designs in the hull of a ship based on standard Navy design criteria. The output is of a general nature and requires interpretation. Presently under development is the interface of HULSTRX and SSDP. The interface program, when completed in early 1983, will allow most longitudinal structure to be input from a design tool to *a* graphic tool without a great deal of manual interface. The engineer still makes all the engineering decisions. The program will work both when a design is near completion and all openings or holes are known, as well as from HULSTRX, when data will be fed to SSDP and all factors of safety can be checked. Under all present design methods, this is only selectively done. Planned FY 83 developments will expand the use of the program with the development of a neutral finite element file from HULSTRX DGL. The mesh may require refinement but the basic structure will be available in a format that will allow expansion as required-and eliminate the need for a great deal of mundane development work. Another interface is a program similar to SSDP that will do commercial design hulls that are required by the Navy. The need to interface all aspects of conventional shis is critical as ninety-five (95) percent of shipbuilding dollars *are* in this area. The idea of making a program that will do special ships in all materials is ideal but not practical. The unique loading and materials used in Hydrofoils, LCAC, SES, and Minehunters, are not needed in a standard program.

Ship Structural File - Program Design

This section describes the design considerations and their implementation in the making of the interactive Ship Scantling File (SSF). In addition, some general thoughts on the trend that is being persued in this kind of work are outlined.

The SSF is essentially a file containing the structural data that can be edited interactively. As a subsequent step, the data in the SSF will, together with the data in the DGL file, be used for the generation of structural drawings for a preliminary and contract design.

The starting point for the SSF is an existing Design Geometry Library (DGL) file. A separate program will make a skeleton SSF based on the structural members in the DGL file. The subsequent editing of this SSF is accomplished by means of programming that enables a user to insert, modify and delete data for structural members represented in the files, as well as add new members.

The structural parts are grouped together as surfaces. Some of the data entered into the file applies only to individual surfaces; some apply to whole surfaces as such.

Apart from the programming that performs the editing functions there are a few utility-type functions that are used for purposes such as starting a new SSF, packing a SSF, creating hard copies of the contents of an SSF, creating shapes file and transfer of data between different types of computers.

Program Design

The major considerations that governed the design and coding effort of this programming development were:

1. Interactivity and user friendliness. Computer programs have frequently been difficult to use. Some computer experience was usually required in order to make runs for the purpose of performing many engineering-type calculations by means of a computer program. The SSF can be edited by personnel without any computer experience. It has also been designed for interactive use and the results will be right there for the user to see.

   The editing program and some of the utility programs are command driven. As much error checking as possible is done when commands and associated data are keyed in. Unacceptable commands and data are flagged immediately and the user can re-enter the proper input. Help functions are also available. Any command may also be terminated by simply entering a $ in the command string.

2. Program transportability. The Navy uses a number of different computers and conversion problems were sought avoided to the greatest extent possible. The development was done on a PRIME computer and then put up on a VAX 11/780.

   The machine-dependent functions such as reading time and date from the computer system as well as some file accessing were isolated in a few subroutines. In addition the initialization of logical unit numbers was all done in one place. As a result of this effort the conversion to the VAX was accomplished without any difficulties.

   Another aspect of transportabilty is the way in which the files used during a run are opened and closed. Both PRIME and VAX have a Command Processor Language which handles that aspect of data management. In order to make the use of the programs as easy as possible, the files could have been opened and closed from within the programs during the run time. Such an arrangement would, however, have required machine-dependent FORTRAN coding.

3. Rapid file access. This aspect is really an extension to the user friendliness considerations, as sluggish response due to slow file access will frustrate users and make the program less acceptable.

With this requirement in mind the file access had to be random rather than sequential. However, as a user would deal with one surface at a time, it was decided to lay the file out such that it would be searched for specific items, first by surface and then by structural component within the surface. This *arrangement* made the programming much simpler than would be the case if a file organization with a hashing mechanism for placing and finding records had been implemented.

4. Since a number of structural members on any one surface will be equal with respect to shape and size, or shape only, the ability to copy data from one member onto the record of another was considered to be important. Consequently, data already entered for one member or for a whole surface can be transferred with ease to another member or surface.

5. Structural members will in most cases be cut from standard shapes. There are presently about 300 such standard shapes available from steel mills in the U.S. A file with such shapes has therefore been made, together with the necessary programming to access this file so that the user only need specify a shape number. The abbreviated description of the shape together with web and flange dimensions will be fetched out of this shapes file and stored with the structural member.

   In some instances special shapes or built-up members will have to be used. The file with standard shapes can be expanded to accomodate such data as well so that a designer will have as easy access to such data as he has to data describing standard steel mill shapes.

6. Protection of data, from unauthorized use as well as accidental destruction, was yet another concern. The data files can therefore be password protected. Whenever opening a datafile for working the user will be asked if a copy of the file to be edited is wanted so that if a run is terminated improperly only that work done since the beginning of the run will be lost.

   In fact, whenever writing a record to the file, new or updated, administrative data pertaining to the surface being worked on as well as the whole file as such is updated. The chances of losing any data due to a computer crash are very small. If the option to have the file copied at the beginning of the run is exercised very little would be *lost*.

Utility Functions

A number of Utility functions were provided with this programming package.

The starting point for a Ship Scantling File is an existing DGL file as generated by the HULSTRX program. Such a file contains all geometrical data for a ship, or part of a ship.

A stand-alone mainline program, SFNEW, will scan a DGL file and create a skeleton DDF. Every surface within the DGL file will be represented on the SSF and within every surface all structural traces will be represented' on the SSF. A trace may result in several structural pieces as the scantlings may change along a longitudinal or a frame. The structural traces are recognized by their names. On a DGL file there will be geometrical information that does not represent structures. Such data will not be represented in the SSF.

Plates will also be represented in the SSF. In a sence a trace can be likened to a plate strake. Such a strake can then be broken down into several individual plates.

SFNEW need only be run once for every SSF file. It generates records for all surfaces and for each structural trace it finds. In addition there will be one plate record for each surface. No specifics as regards structural data will be associated with the surfaces and the structural parts. In a sense, all records will be empty.

By initializing an SSF this way, a lot of work is saved in creating the records. This process lays out the file. Deletions and additions can be done at any time later on.

Once a SSF has been established, it will be necessary to manage it. The functions that are required are checking of password, change of the password, packing the file and producing hard copies of it. The program that does this is named SFUTIL and is command driven just as the Editing program itself. Only the person, or persons, that are authorized to perform file management functions are supposed to have access to this program.

The library of standard shapes that will be referenced can be updated by means of a stand-alone program named SHAPES. The main function of this program is to expand the file of standard shapes *to* include Special and built up shapes. It is also to be used for the purpose of modifying *or* removing special shapes. SHAPES is command driven, just as SFNEW. It is not expected that SHAPES will be used very often.

Apart from these utility functions, there are two additional programs that *exist* for the purpose of moving data for the DGL and SHAPES files between two computer systems. All files used by any program are binary, i.e. no data conversion from internal representation according to a FORMAT statement occurs when reading from a *file* or the other way around, when writing to a file. Thereofore, in order to transfer files between computer systems the binary files are converted to data in ASCII format. *Files* on the system from which the transfer is to take place will be converted from binary to ASCII and then transferred via tape. On the receiving system such ASCII files will be converted back to binary.

The program that converts data from a DGL file was made for the HULSTRX program and only put up on the PRIME. One minor modification had to be made inorder to make this program work on both PRIME and VAX computers. This program performs the conversions both ways, i.e. either from ASCII to binary or the other way around.

A similar program was made for the purpose of transferring data for the SHAPES.

Editing Program

The main editing progran is named SFEDIT. It is programmed in FORTRAN and makes use of system subroutines for reading time and date as well as accessing the SSF. The reason for using a system subroutine for file read and write rather than standard FORTRAN random access read and write statements is that processing on the PRIME is much faster by using the system subroutine. However, it is perfectly possible to use the FORTRAN statements. The VAX implementation was done that way. It is not possible to use both access methods on a PRIME file as there is a little difference in the way the data is laid out.

Two concepts need be explained before the editing functions are outlined:

Current The record or records being worked on are the current record or records. If a surface as such is being worked on, supplying new, replacing existing data or deleting data, then the surface is current. All functions relate to the surface. If the surface is deleted, i.e. removed from the file, then all records for plates or structural pieces belonging to the surface as well the surface record is removed.

When a plate or structural part is current, then both the surface and the plate or piece are current. However, when such a plate or piece is removed the surface and all other plates and pieces under it remain as is.

Active Data describing the surface and its individual plates or pieces are entered by means of some of the commands described below and placed in a buffer. The data in this buffer is referred to as the active data. Whenever a record is filed or stored data from the active buffer is written out to the record on the file.

The active buffer may contain data pertaining to *a* surface, to a plate and to a structural piece. During the filing operation only that which pertains to the type of record is written to the record on the file.

The editing functions that *are* provided in the editing programs are these:

BOUND Prompts for bounding data for a surface or plate or structural piece.

CLEAR Clears all data from the active buffer.

DELETE Deletes the current surface with all associated plate or structural pieces or individual pieces only.

END Terminates the editing session.

FILE Writes active data on current record.

GET Reads specified record from the SSF.

IDNAME Lists standard 6-character identifiers.

INTERSECT Prompts for intersection data for the current surface.

LIST Provides listing of specified type of records which may include all surfaces in the file or all plates or structural pieces within the current surface.

MATERIAL Prompts for the material type.

NEXT Displays next logical record on SSF.

ORIENT Used to define orientation.

PIECE Finds and displays specified piece.

PLATE Finds and displays specified plate.

SCANTLING Prompts for plate/shape scantling.

SET Makes parameters of current SSF record active, i.e. moves data from file record to active buffer.

SHAPE Finds and displays specified shape.

STATUS Displays non-zero active data for the kind of record being current.

SURFACE Finds and displays specified surface.

All the commands can be abbreviated to the first two letters, i.e. FI for FILE, MA for MATERIAL, PI for piece, etc.

Most of the commands will read the associated data from the same line as the command and will prompt the user for that which is not found. Some commands will prompt for individual data as it would not all fit on one line.

There is also a HELP function which will list all the commands with a brief description of what they do. If a command is not recognized the user will be given -a message to that effect.

All data read will be interpreted by the program rather than a FORTRAN FORMAT statement which may cause program termination if the data is mistyped. By having it this way, errors made in typing can be corrected by retyping.

Future Data Base Design - Considerations

This section outlines the requirements of a future data base based on the present experiences.

The present state of the programs used in ship design and manufacturing in the U.S. and indeed around the world is that interactivity such as is now possible with multi-user operating systems on very reasonable mini-computers is not fully taken advantage of. A number of programs that where written in the days of cards and batch processing are still around.

Special interfaces have been and are being made to allow output from one program to be used as input to another. Many existing programs are difficult to use due to input requirements designed for cards with little or no concern for other uses, or for how the input could be made flexible. Another way of expressing this would be to call it user unfriendly.

Transfer of data between different types of computer equipment is frequently less easy than it is possible to make it. This applies in particular to systems with reasonably sophisticated data base designs. In particular, it is not possible to down-load portions of a data base to a smaller computer over a dial-up communication link for off line work and then to transfer the results of such work to a larger data base over the same dial up communication link at a later stage.

Another aspect of present day software technology is that some programs have to access different data files in different ways in order to get at all the data that is needed for a specific application.

Significant improvements in the ease of use of programs can be made with a relatively small effort towards making existing programs truly interactive and in designing data base systems that will enable transfer of data between different application programs and between different computer equipment. The rest of this section is devoted to some of the design requirements of such a data base.

The main requirements of such a data base are as follows:

Files — Sequential files are so slow to access in any other way than record by record as they occur that the only acceptable way is random access. Such an access will in most instances require a hashing mechanism, although it is possible to design access methods that will use data from a previous record to calculate the position of the next one to be examined.

A hashing mechanism imposes a software overhead and will require more physical disk space than a sequential file due to the fact that some record spaces will not be filled. However, it is the best way to find where to put a record and where to find it again later in as short a time as possible. The response from the system is extremely important when a file is being accessed interactively.

There are other ways of placing records in a direct access file than hashing, but the programming overheads are usually as large and the response would be slower because it would involve some searching in a register.

Records

The records to be used must necessarily be of varying length and they must not be laid out in just one fixed way. It is possible to store one or a couple of keys with the record that can be used *to* indicate both lay-out and length.

The requirement of having records of varying lengths makes it necessary to have an index area which must remain in the file at all times. Each record will have its name entered here together with information such as time/date stamp and pointers to where the record data is stored. As records are entered, the file will grow. Since the records are to be of varying lengths, they must consist of a number of blocks, each of standard length. A record will therefore consist of one or more such blocks apart from the entry in the file register.

The position of a record in the register will be determined by a hashing mechanism. If all records were to be of one size, then the file could be initialized with a number of such record. The name, date/time stamp and other administrative data could reside in the record together with the actual record data.

Access

There are two aspects of access. It should be easy to store and retrieve data in the files for just about any program and it should be possible to transfer records between computer systems over dial-up communication lines. In particular, it will become important to be able to down-load only a part of a data base to a smaller stand alone computer and then to up-load the result of work done on such a system to the host computer.

There are far too many programs today that still operate independently of related programs. Programs written for the computer equipment of yesterday have been converted to read from a file rather than a card reader and to write to a file instead of a line printer. Such programs were made with little or no concern for other uses of the output. Some were made to produce a deck of cards, or a file that could be used by other programs.

The more advanced systems such as CAD/CAM programs were made with highly specialized data bases that were difficult, if not impossible, to access for uses other than those associated with the system for which it was made.

Interactivity In going from batch oriented programming systems to interactive systems, the old databases were frequently retained or slightly modified. The time in accessing such data bases frequently were too large for the system to be accepted by users. Rapid response from an interactive system will depend on the data base design as far as retrieval of data is concerned. Sluggishness in such responses will make a system less acceptable.

The second aspect of interactivity is the way in which a system lets users enter data. This has no direct bearing on the data base design, but deserves a few words in this context. Well designed screen layouts with prompts can be very helpful. Error checking at the earliest possible stage is also important in that it lets the user recover as early and as often as possible.

Sharing Data Down-loading parts of a data base to a stand-alone system will become more desirable as the hosts themselves become smaller and smaller. A number of functions can be performed by stand alone micros which can go in and fetch data for a day's work. At the end of the day the result can be put back into the databank in the host. Over a dial-up line, independent work stations can be set up in a different location and reduce the demand on CPU in a central computer. At the end of a task the result can then be transmitted back to the host.

The result of work in an independant workstation may result in-additional records or modification of those that were down-loaded. The modified ones will replace the original ones when the result is sent back into the host computer.

Another reason for communications is that a data bank or part of one may be transferred between computers, thus avoiding delays associated with sending tapes or floppy disks through the mail.

The communication interface at a host computer can be written as an extension to the data base programming. It will be required to fetch a specified record from a file, possibly do some conversions and then transmit bytes down the line. The records moved that way will be put into a data base in the receiving system. The specific file into which they will go need not be the same size as that in the sending system. When a record is entered into any such system, its name is hashed and the result of that hashing will be a fraction as well as a specific number for the file into which it is to go first. For subsequent transfers, the fraction will be used to determine the place in any data base.

Portability — The software needed to implement a scheme such as has been outlined above needs to be portable if it is to be implemented on several types of computers. Efficiency will, however, demand that some parts will be accomplished by means of system subroutines already available for some systems or by assembly on others. The important portability aspect here is well-defined specifications so that certain pieces of software such as file handling, file access input routines and communications can be implemented on a number of different types of computers -- micros, super-micros, minis and mainframes.

Data Safety — A data base system to be put in place on several physical computers and linking these by telephone lines must necessarily have built-in safe guards against the loss of data whenever a run is terminated improperly due to a line being lost. There are several design features that can be built in to minimize the loss and to make recovery as painless as possible. Just to indicate the kind of features that might be designed in it can be mentioned that a continual updating of all administrative data on the physical disk file will minimize the loss. A recovery can be made fairly easy by keeping one record that at all times indicates the very last record put in place and some data regarding it.

Security — By making a data base accessible from several locations it will be necessary to provide some security in the form of passwords and scrambling. These measures are meant only to make it difficult to access data that should not be available to just anybody. They are not meant to protect classified data. However, passwords and scrambling can be a considerable deterrent to anyone attempting to see what kind of data is in a file.

References

(1) Thomas P. Gallagher, REDUCING PRODUCTION MANHOURS THROUGH DESIGN OFFICE PROCEDURES, STRUCTURAL DESIGNER-FABRICATOR RELATIONSHIP, REAPS Technical Symposium, 1978.

(2) Stephen Klomparens, HULSTRX - A CASDAC Computer Aid for Hull Structural Contract Design
Proceedings of the REAPS Technical Symposium, 1979.

(3) Eric Byler and David Helgerson, U.S. NAVY CAD/CAM PROGRAM HULSTRX
Proceedings of the REAPS Technical Symposium, 1981.

(4) Eric Byler and Ronald Walz, U.S. NAVY CAD/CAM PROGRAM FOR HULL STRUCTURE (HULSTRX)
Proceedings of the ICCAS Technical Symposium, 1982.

(5) Properties of Steel Shapes and Plate-Beam Combinations used in Shipbuilding, Military (Navy) Handbook, September 1979.

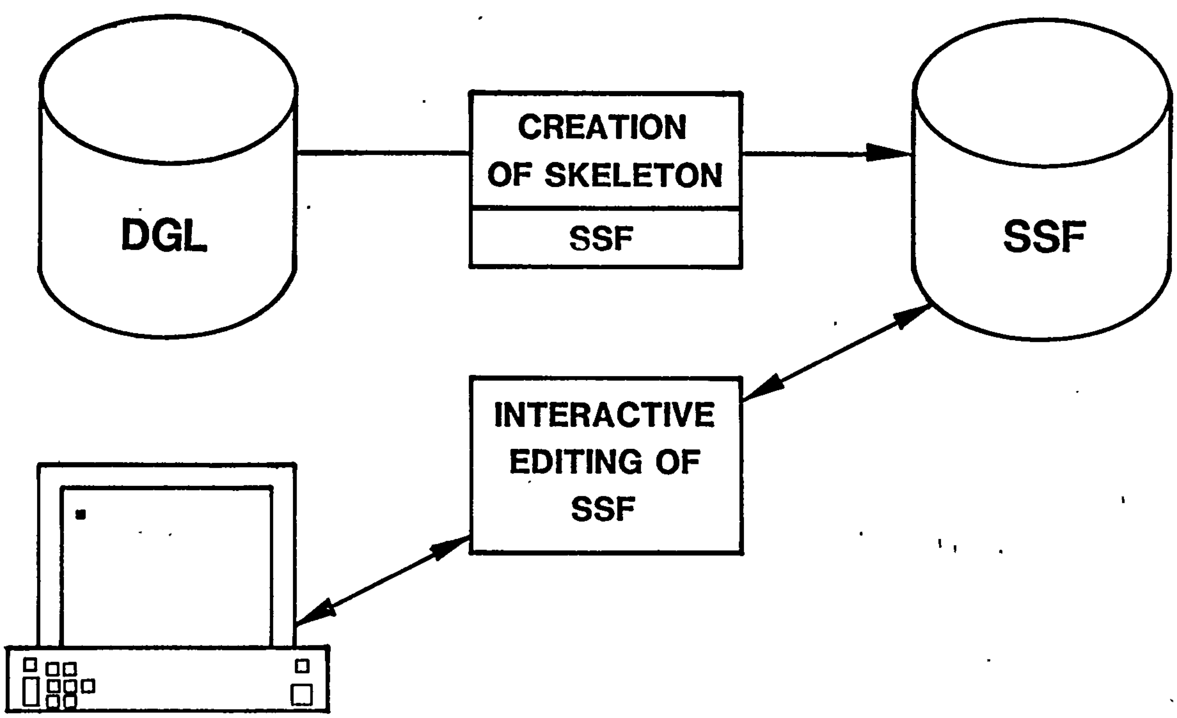


Figure 1 -Creation of SSF

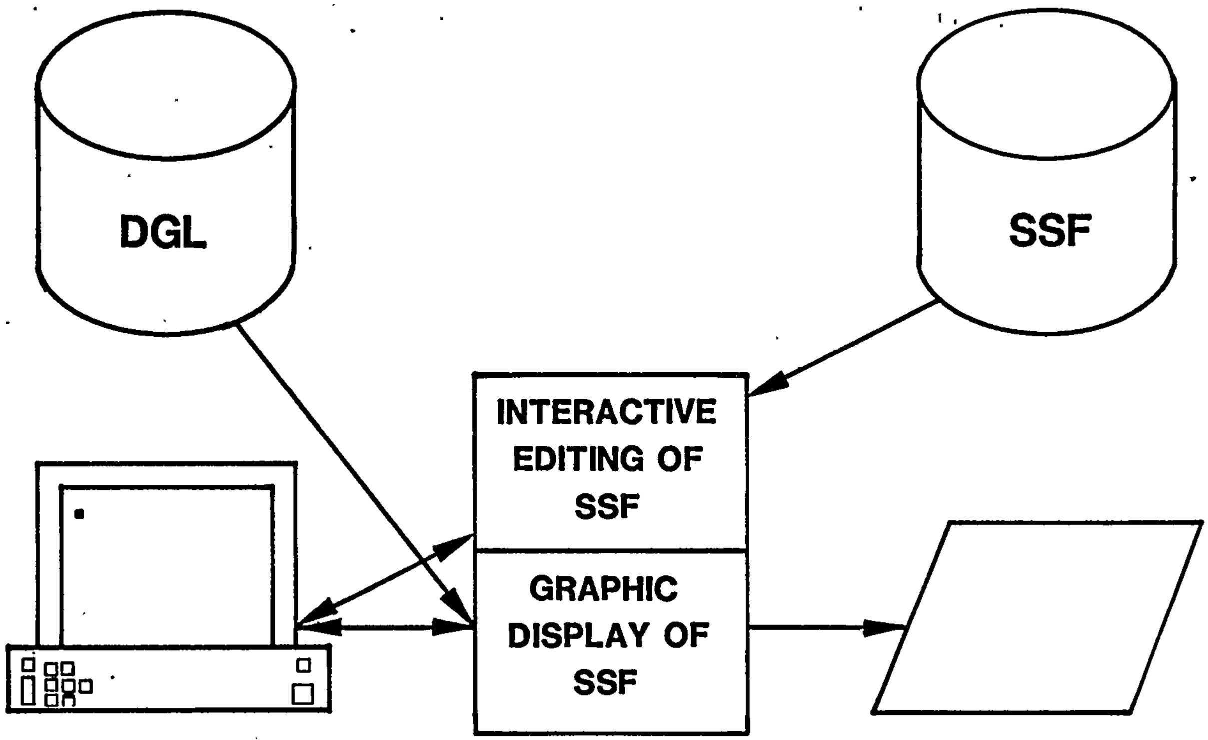


Figure 2 - Graphic Display of SSF

SSF PROGRAM DESIGN CONSIDERATION

- INTERACTIVITY
- PORTABILITY
- RAPID FILE ACCESS
- EASY FILE EXPANSION
- PROTECTION OF DATA

SSF UTILITY FUNCTIONS

- CREATION OF SKELETON FILE
- CREATION OF STANDARD SHAPES FILE
- PASSWORD CHANGES
- PACKING
- TRANSFER OF DATA BETWEEN COMPUTER SYSTEMS

SSF EDITING FUNCTION/DEFINING STRUCTURES

- BOUNDING STRUCTURES AND SURFACES
- DEFINING SCANTLINGS OF SHAPES
- DEFINING PLATE THICKNESS OF PLATES
- DEFINING MATERIAL
- ORIENTATION
- DEFINING SURFACE INTERSECTION

SSF EDITING FUNCTIONS/SEARCHING

- FINDS ANY MEMBER ON FILE
- FINDS NEXT PLATE. OR- PIECE
- FINDS SURFACE
- FINDS SPECIFIED SHAPE WITHIN SURFACE
- FINDS SPECIFIED PIECE WITHIN SURFACE
- FINDS SPECIFIED PLATE WITHIN SURFACE

ANY SURFACE, PLATE OR PIECE NOT FOUND RESULTS IN CREATION OF NEW ITEM

SSF EDITING FUNCTION/DATA MOVEMENTS

- SETS ACTIVE BUFFER EQUAL TO SCANTLINGS FOR SURFACE, PLATE OR PIECE
- FILES SCANTLING RECORD, SURFACE PLATE OR PIECE
- DELETE SCANTLING RECORD FOR SURFACE PLATE OR PIECE

REQUIREMENTS FOR FUTURE DATA BASE

- RAPID FILE ACCESS
- RECORD LENGTHS OF VARYING LENGTHS
- DIAL UP ACCESS
- *EASY ACCESS FOR ANY PROGRAM
- INTERACTIVlTY
- SHARING DATA BETWEEN COMPUTERS
- PORTABILITY
- DATA SAFETY
- DATA SECURITY

Additional copies of this report can be obtained from the
National Shipbuilding Research and Documentation Center:

http://www.nsnet.com/docctr/

Documentation Center
The University of Michigan
Transportation Research Institute
Marine Systems Division
2901 Baxter Road
Ann Arbor, MI 48109-2150

Phone: 734-763-2465
Fax: 734-763-4862
E-mail: Doc.Center@umich.edu